Towards Improving the Security of Low-Interaction Honeypots: Insights from a Comparative Analysis

Abubakar Zakari*[1], Abdulmalik Ahmad Lawan*[2], Girish Bekaroo[3]

[1,2]Department of Computer Science, Kano University of Science and Technology Wudil, Kano Nigeria
[3] School of Science and Technology, Middlesex University (Mauritius Branch Campus), Vacoas, Mauritius
{*[1] abubakar.zakari@yahoo.com; *[2] aaltofa2000@gmail.com; [3]g.bekaroo@mdx.ac.mu}

Abstract. The recent increase in the number of security attacks by cyber-criminals on small businesses meant that security remained a concern for such organizations. In many such cases, detecting the attackers remained a challenge. A common tool to augment existing attack detection mechanisms within networks involves the use of honeypot systems. A fundamental feature of low-interaction honeypots is to be able to lure intruders, but the effectiveness of such systems has nevertheless been affected by various constraints. To be able to secure honeypots systems, it is important to firstly determine its requirements, before taking appropriate actions to ensure that the identified requirements have been achieved. This paper critically examines how existing low-interaction honeypot systems abide to major requirements before recommending how their security could be improved.

Keywords: Low-Interaction Honeypots, Deception in Depth (DID), Deceptiveness, Intelligence, Robustness.

1 Introduction

Even though the massive growth of the internet over the past years provided various benefits to end users and businesses, security remained a concern [1, 2]. It has recently been reported that there has been a huge rise in attacks as cyber-criminals have been targeting small businesses [3]. In many such attacks, detection of the attackers remained a challenge. Although security systems like intrusion detection systems (IDS), firewalls, intrusion prevention systems (IPS) have been existent since many years to enhance the security of networks; various issues were raised with regards to detection of new attacks [4]. A common tool to augment existing attack detection mechanisms within networks is honeypot and by using such systems, new attacks could be uncovered, assault patterns might be revealed, and the precise thought processes of the intruder could be studied [5].

A key purpose of a honeypot is to serve as a decoy used to lure intruders in order to accumulate important information about the intruder and technique of attack that was used to compromise the system. The gathered information could then be used by the

organization to trace back the attacker and to also improve its internal defense mechanisms. Honeypot systems can be developed for two reasons purposes, namely, production and research, and can either be of low-interaction or high-interaction [6]. Low-interaction honeypots simulate some portion of the operating system for instance the network stack, while focusing on services that cannot be utilized by the intruder to adventure the real system. This type of honeypot normally implements only the Internet protocols to permit interaction with intruder while making the latter believe the real system is being compromised [7]. On the other hand, high interaction honeypots are complete production similar systems that have a full set of services and permit an intruder a great deal of scope throughout the intrusion. Generally, high interaction honeypots are challenging to recognize and are costly to maintain [8]. The costs of high interaction honeypots is a barrier to their adoption by small businesses and also most firms do not need high-interaction honeypots that captures massive amounts of data [9].

A fundamental feature of low-interaction honeypots is to be able to lure intruders, but the effectiveness of such systems has nevertheless been affected by various constraints. In the past, attackers have been using OS fingerprint techniques such as NMAP, Xprove to remotely attack and distinguish honeypot from a real system [10]. Moreover, data security experts have been increasing focus on defensive strategies while neglecting offensive strategies [6] thereby increasing its vulnerability to intruders. If a honeypot is discovered by an intruder, its purpose is defeated and no advantage is provided to the organization adopting it. As such, improvement of existing honeypot systems is needed. To be able to secure network systems in general, it is important to firstly determine its requirements, before taking appropriate actions to ensure that the identified requirements have been achieved [11].

In terms of related work, although various studies focused on improving honeypot systems against newly identified vulnerabilities and attacks, limited published literature is available on the comparison related to how existing low-interaction honeypots adhere to their security requirements. As such, this paper critically examines how existing low-interaction honeypot systems abide to key identified requirements before recommending how their security could be improved. The analysis and recommendations provided in this paper could be used by researchers and experts in their endeavor to improve the security of low-interaction honeypot systems to eventually benefit businesses. The paper is structured in the following manner: In section 2, the security requirements of honeypots are investigated before reviewing the existing honeypot systems in section 3. Results from a comparative analysis is provided in section 4 followed by recommendations on how the security of low-interaction honeypots could be improved in section 5. Finally, the work is concluded in section 6.

2 Key Requirements of Honeypots

A major security requirement of honeypot systems is its deceptiveness [12, 13]. Deceptiveness involves obscuring valuable data in bland-looking files, and set up honeypots that divert attackers from the real assets whereby leading them to false

intellectual property, or causing them to trip alarms [14, 15]. In short, deception involves misleading attacker into believing something that is false. Among the deceptive techniques, camouflage involves disguising the network infrastructure by making it a moving target, changing addresses, infrastructure topologies, and available resources daily. In other words, camouflaging take steps to prevent attackers from seeing the same infrastructure twice [16]. Disinformation is another process which involves diverting or confusing attackers with false information [15]. In this process as well the hacker is supplied with fake successes, responses, files, and assets to exploit. Also, disinformation poised that any false information given must not be easily disprovable. Moreover, work has also been done to categorize the sophistication of deceptive discipline into different levels, namely, static, dynamic and adaptive deception [17]. Static deception has been referred as constant execution of an often uncontrollable trait whereas dynamic deception is implemented upon activated response to some stimuli. Adaptive deception in turn adjusts and reacts to a situation while also and employing cognitive assessment before, during and after the fact.

Another essential requirement of honeypots systems are their robustness and fault tolerance [7, 18]. A system is said to be fault-tolerant if it is able to automatically recover from errors or faults while also being able to eradicate faults without suffering from an externally perceivable failure [19]. This essential ability ensures that honeypots are able to recuperate while at the same time guarantees robustness as the system is able to also cope with errors during execution [7].

Furthermore, intelligence of honeypot systems is also important. Intelligence enables honeypots to gain actionable insights by gathering threat intelligence feeds and adversary indicators that define and describe trends, tendencies, methods, and actions taken by attackers [20]. Intelligent honeypot pretend to surrender to one form of attack in order to suppress a second, less-obvious defense. With series of attacks on different levels of relevance and context, intelligent honeypot continue to ramp up their threat intelligence capabilities while increasing their effectiveness with regards to intelligence-led deceptions.

3 Low-Interaction Honeypot Systems

Different low-interaction honeypot systems have been proposed and this section reviews the common ones.

3.1 Honeyahole

Principally designed and developed to escape from honeypot hunting, honeyahole implements three phases, namely, collection, redirection and deception in order to gather four types of attacking information to build up the blacklist [6]. The honeypot has two redirection techniques embedded to dynamically send incoming traffic to a production or a deception server in the same redirection phase.

3.2 Honeywall

Honeywall is a honeypot that helps in deploying honeynet with ease by automating the process of deployment [2]. This honeypot can also capture and analyze traffic (both inbound and outbound) of honeynet activity. An identified vulnerability of this honeypot involves construction of a traffic stream that consists of strings matching snort_inline's rewriting database before verifying whether all packets are received unmodified [21].

3.3 Honeyd

Honeyd is a low-interaction, open source honeypot, and can be deployed in various platforms (Windows, UNIX) [22]. It can emulate operating systems at TCP/IP stack level and also monitor all UDP and TCP based ports (as shown in Fig. 1). A few vulnerabilities of this type of honeypot were also found. Studies showed that Honeyd can be detected remotely using fingerprint attacks [23, 24] and using timing analysis of ICMP ECHO request [5]. In an attempt to improve the identified limitations, work has been done to create a new camouflaged Honeyd by modifying the original honeypot in addition to the underlying operating system support in order to permit high-fidelity emulation of events [25].

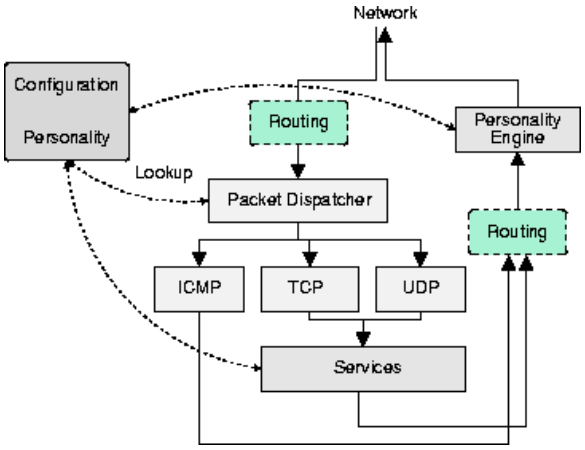

Fig. 1 Honeyd infrastructure [25]

3.4 Honeytrap

Being a low-interaction honeypot that operates by observing attacks against network services, honeytrap aims at collecting malware in an automated manner [14]. It permits the collection of traffic information for pre-closed ports by opening them in which an access is observed. As a limitation, this honeypot is not able to capture

details pertaining to the activities of an attacker unless a second attempt is made to the same honeypot [26].

3.5 Nepenthes

As a honeypot used for malware collection, Nepenthes inherits the scalability of low-interaction honeypots and its flexibility enables it to emulate vulnerabilities of different operating systems on a single machine during a single attack [27]. In terms of limitations, Nepenthes is only capable to collect autonomously propagating malware and that malicious software which spread using hitlist to find vulnerable systems are hard to detect [28]. Furthermore, since Nepenthes emulates huge number of vulnerabilities, it makes it easy to detect by attacker, because many TCP ports are open in the process [28].

4 Comparative Analysis

Literature analysis reveals that the focus of the different reviewed low-interaction honeypots vary in terms of characteristics and abilities. For Honeyd, detection using OS fingerprinting attacks reduces its deceptiveness, camouflaging and robustness capabilities [23, 24], although it has the ability to emulate sensitive features of operating system and gather vital information of attacker as well, thus illustrating intelligence [24]. On the other hand, Honeywall depicts robustness and intelligence due to its ability to capture and analyze traffic (both inbound and out bound) of honeynet activity [20]. Moreover, this honeypot has also been portrayed as intelligent as it is able to deceive and link the attacker to the honeypot system [21]. Honeytrap, in turn, does not collect details of attacker on the first attempt the attack is made but rather relies on further attack attempts to be able to correctly detect the attacker which is a big disadvantage in terms of intelligence [26]. Honeyahole is basically designed to escape honeypot hunting with focus on camouflaging, deceptiveness and robustness [6]. Finally, although Nepenthes is portrayed as a highly deceptive honeypot, it has different vulnerabilities thus making it easy to detect by an attacker [28]. Moreover, due to its inability to detect malwares that propagates using a hitlist makes its intelligence and robustness undermined [28]. Comparisons are summarized in Table 1 to show how existing low-interaction honeypot systems have been portrayed by literature.

Table 1. Comparative Analysis

Item	Deceptiveness	Robustness	Intelligence
Honeyd	√	×	√
Honeywall	√	√	√
Honeytrap	√	√	×
Honeyanole	√	√	×
Nepenthes	√	×	×

From Table 1, it could be seen that deceptiveness has been the major focus of the reviewed low-interaction honeypots towards misleading attackers. The five honeypots showed to have this capability although a few vulnerabilities have been highlighted especially relating to the use of Honeyd and Nepenthes. Robustness, which relates to coping with errors during execution, was identified as an important requirement of honeypots. However, literature showed that this requirement has not been a major focus of a few honeypots. Among the three requirements, intelligence seems to be the least focused aspect of low-interaction honeypots. Among the 5 honeypots that were compared, 3 of them did not portray intelligence abilities, which is important to better deceive attackers while also accurately obtaining their details.

5 Recommendations

Results showed that the key requirements needing attention are robustness and intelligence of low-interaction honeypot systems. Robustness can be improved using redundancy or collaborative honeypot systems such that in case one of them fails, others remain operational. Faults within existing honeypots could also be isolated by further testing such systems. On the other hand, different works have been conducted to improve the intelligence of low-interaction honeypots. First of all, honeypot systems can embed intelligence by learning the moves of attacker in addition to tools used to compromise systems [29]. Also, honeypots systems can be made dynamic whereby having the capability to learn about network environments and infrastructures before autonomously deploying individual honeypots based on current layout [20]. Furthermore, after deployment, such systems should be able to repeatedly monitor network changes and update configurations accordingly [20]. Additionally, the Deception-in-Depth (DiD) concept of operation could be utilized [30]. DiD utilizes the layering approach with three different layers in the proposed model aimed at strengthening the defense of honeypot systems [31]. Within the model, the honeypot asset is represented in the innermost layer whereas the honeypot is positioned in the middle layer of the model. The purpose of the outermost layer is to improve deception using techniques including fake access points.

6 Conclusions

This paper examined how existing low-interaction honeypot systems abide to their requirements before recommending how their security could be improved. Three important requirements of honeypot systems were identified namely deceptiveness, robustness and intelligence. Among these requirements, existing low-interaction honeypots seem to focus on deceptiveness with reduced attention given to their robustness and intelligence. As such, more work is needed towards improving robustness and intelligence of low-interaction honeypot systems so as to improve their overall effectiveness. As future work, the proposed recommendations could be practically evaluated to assess their effectiveness. Moreover, a framework could be proposed focusing on the three requirements investigated in this study to help

businesses in their endeavor to prevent attackers from detecting, exploiting, and deceiving honeypot systems and assets.

7 References

1. Chakrabarti, A., Manimaran, G.: Internet infrastructure security: A taxonomy. IEEE network. vol. 16. no. 6 (2002) pp. 13-21.
2. Tiwari, R., Jain, A.: Design and analysis of distributed honeypot system. International Journal of Computer Applications. vol. 55. no. 13 (2012) pp. 20-23.
3. Smith, M.: Huge rise in hack attacks as cyber-criminals target small businesses. [Online] (2016) Available at: https://www.theguardian.com/small-business-network/2016/feb/08/huge-rise-hack-attacks-cyber-criminals-target-small-businesses [Accessed 20 Aug 2016].
4. Yang, Y., Yang, H., Mi, J.: Design of distributed honeypot system based on intrusion tracking. In IEEE 3rd International Conference on Communication Software and Networks (ICCSN), pp. 196-198, IEEE (2011).
5. Mukkamala, S., Yendrapalli, K., Basnet, R., Shankarapani, M.K., Sung, A.H.: Detection of virtual environments and low interaction honeypots. In Information Assurance and Security Workshop, 2007. IAW'07. IEEE SMC, pp. 92-98. IEEE (2007).
6. Shiue, L., Kao, S.: Countermeasure for detection of honeypot deployment. In IEEE International Conference on Computer and Communication Engineering 2008. ICCCE 2008. pp. 595-599. IEEE (2008).
7. Mohammadi, S., Nikkhahan, B.: A fault tolerance honeypots network for securing E-government. In IEEE International e-Conference on Advanced Science and Technology, 2009. AST'09. pp. 13-17. IEEE (2009).
8. Defibaugh-Chavez, P., Veeraghattam, R., Kannappa, M., Mukkamala, S., Sung, A.H.: Network based detection of virtual environments and low interaction honeypots. In 2006 IEEE Information Assurance Workshop, pp. 283-289. IEEE (2006).
9. Brown, B.: How to make a honeypot network security system pay off. [Online] (2007) Available at: http://www.networkworld.com/article/2296754/lan-wan/how-to-make-a-honeypot-network-security-system-pay-off.html [Accessed 10 Aug 2016].
10. Valli, C.: Honeyd-A OS Fingerprinting Artifice. In Proceedings of 1st Australian Computer Network and Information Forensics Conference (2003).
11. Bishop, M.: What is computer security?. IEEE Security & Privacy. vol. 1. no. 1 (2003) pp. 67-69.
12. Cohen, F.: The use of deception techniques: Honeypots and decoys. Handbook of Information Security. vol. 3. no. 1 (2006) pp. 646-655.
13. Zhang, F., Zhou, S., Qin, Z., Liu, J.: Honeypot: a supplemented active defense system for network security. In Proceedings of the Fourth International Conference on Parallel and Distributed Computing, Applications and Technologies 2003. PDCAT'2003. pp. 231-235. (2003).
14. Provos, N.: A Virtual Honeypot Framework. USENIX Security Symposium, vol. 173 (2004) pp. 1-14.
15. Rowe, N.: Deception in Defense of Computer Systems from Cyber Attack. Cyber Warfare and Cyber Terrorism (2008).
16. Fu, X., Yu, W., Cheng, D., Tan, X., Streff, K., Graham, S.: On recognizing virtual honeypots and countermeasures. In 2nd IEEE International Symposium on Dependable, Autonomic and Secure Computing 2006. pp. 211-218. IEEE (2006).
17. Yek, S., Australia, W.: Measuring the Effectiveness of Deception in a Wireless Honeypot. In Australian Computer, Network & Information Forensics Conference (2003).
18. Nikkhahan, B., Aghdam, A., Sohrabi, S.: E-government security: A honeynet approach. International Journal of Advanced Science and Technology. vol. 5. (2009).

19. Avizienis, A., Kelly, J.: Fault tolerance by design diversity: Concepts and experiments. Computer. vol. 17, no. 8 (1984) pp. 67-80.
20. Zakaria, W., Kiah, M.: A review on artificial intelligence techniques for developing intelligent honeypot. In 2012 8th International Conference on Computing Technology and Information Management (ICCM), pp. 696-701. IEEE (2012).
21. Provos, N., Holz, T.: Virtual honeypots: from botnet tracking to intrusion detection. Pearson Education (2007).
22. Krutz, R., Vines, R.: The CEH Prep Guide: The Comprehensive Guide To Certified Ethical Hacking (With CD), John Wiley & Sons (2007).
23. NOSTROMO: Techniques in OS-Fingerprinting. Hagenberg (2005).
24. Boyle, A.: A Remote OS Identification Primer. SANS (2001).
25. Fu, X., Graham, B., Cheng, D., Bettati, R. Zhao, W.: Camouflaging virtual honeypots. Texas A&M University (2005).
26. Song, J., Takakura, H., Okabe, Y.: Cooperation of intelligent honeypots to detect unknown malicious codes. In IEEE WOMBAT Workshop on Information Security Threats Data Collection and Sharing, 2008. WISTDCS'08. IEEE (2008).
27. Kumar, S., Sehgal, R., Singh, P., Chaudhary, A.: Nepenthes Honeypots based Botnet Detection. Journal of Advances in Information Technology. vol. 3. no. 4. (2012) pp. 215-221.
28. Baecher, P., Koetter, M., Holz, T., Dornseif, M., Freiling, F.: The nepenthes platform: An efficient approach to collect malware. In International Workshop on Recent Advances in Intrusion Detection (2006).
29. Gupta, N.: Improving the effectiveness of deceptive honeynets through an empirical learning approach. In 3rd Australian Information Warfare & Security Conference (2002).
30. Yek, S.: Implementing network defence using deception in a wireless honeypot. In Australian Computer, Network & Information Forensics Conference (2004).
31. Gerwehr, S., Anderson, R.: Employing deception in INFOSEC. [Online] (2000). Available: http://www.cert.org/research/isw/isw2000/papers/26.pdf. [Accessed 10 Aug 2016].